

**BEST AVAILABLE COPY**

**JP2003087316**

Publication Title:

METHOD OF TRANSMITTING DATA

Abstract:

Abstract of JP2003087316

**PROBLEM TO BE SOLVED:** To provide a method of transmitting data capable of avoiding network congestion. **SOLUTION:** This invention provides the method of transmitting data from customers over a computer network, in particular over the Internet, where the data to be sent are split into packets, in particular into IP packets, where each packet is marked by one of at least two states (IN, OUT) and where the states (IN, OUT) determine which packets are dropped first, if packets are dropped during transmission and the marking of the packet with a state of high drop precedence (OUT) is based on a random probability ( $p$ ).

Data supplied from the esp@cenet database - Worldwide

---

Courtesy of <http://v3.espacenet.com>

(19)日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-87316

(P2003-87316A)

(43)公開日 平成15年3月20日 (2003.3.20)

(51)Int.Cl.  
H 04 L 12/56識別記号  
200F I  
H 04 L 12/56テーマコード(参考)  
200Z 5K030  
200B

## 審査請求 未請求 請求項の数28 O.L 外国語出願 (全 47 頁)

(21)出願番号 特願2002-252727(P2002-252727)

(22)出願日 平成14年8月30日(2002.8.30)

(31)優先権主張番号 10142426.4

(32)優先日 平成13年8月31日(2001.8.31)

(33)優先権主張国 ドイツ(DE)

(31)優先権主張番号 10209705.4

(32)優先日 平成14年3月6日(2002.3.6)

(33)優先権主張国 ドイツ(DE)

(31)優先権主張番号 10220213.3

(32)優先日 平成14年5月6日(2002.5.6)

(33)優先権主張国 ドイツ(DE)

(71)出願人 000004237  
日本電気株式会社  
東京都港区芝五丁目7番1号

(72)発明者 サンドラ タルタレリ  
ドイツ共和国, 69115 ハイデルベルク,  
アデナウアー プラッツ 6, エヌイーシー  
ヨーロッパ リミテッド内

(74)代理人 100071272  
弁理士 後藤 洋介 (外1名)

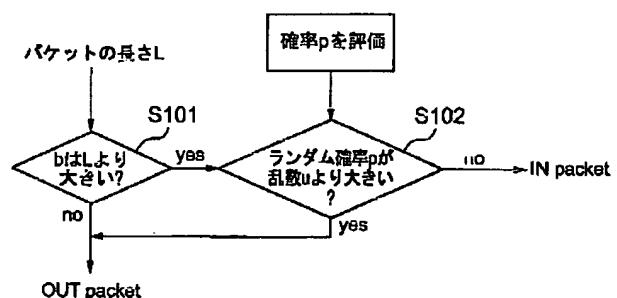
最終頁に続く

## (54)【発明の名称】 データ伝送方法

## (57)【要約】

【課題】 ネットワーク輻輳を回避することができるデータ伝送方法を提供することである。

【解決手段】 コンピュータ・ネットワークを介して、特にインターネットを介して、顧客からデータを伝送する方法において、送信されるべきデータはパケットに、特にIPパケットに、分割され、各パケットには少なくとも2つの状態(イン、アウト)のうちの一つで印が付けられる。その状態(イン、アウト)は、もし伝送中にパケットが落とされるとすればどのパケットが最初に落とされるかを決定し、高廃棄優先順位(アウト)の状態でのパケットの印付けがランダム確率( $p$ )に基づく。



## 【特許請求の範囲】

【請求項1】 コンピュータ・ネットワークを介して、特にインターネットを介して、顧客(C1、C2、C3、C4、C5、C6、C7、C8、C9、C10)からデータを伝送する方法であって、この方法では送信されるべきデータはパケットに、特にIPパケットに分割され、前記各パケットには少なくとも2つの状態(イン、アウト)のうちの一つで印が付けられ、その状態は、もし伝送中にパケットが落とされるとすればどのパケットが最初に落とされるかを決定し、高廃棄優先順位(アウト)の状態でのパケットの印付けがランダム確率(p)に基づくことを特徴とするデータ伝送方法。

【請求項2】 前記高廃棄優先順位(アウト)でのパケットの印付けは単一のランダム確率(p)に基づくことを特徴とする請求項1に記載のデータ伝送方法。

【請求項3】 前記ランダム確率(p)は各顧客(C1、C2、C3、C4、C5、C6、C7、C8、C9、C10)のトラフィックについて測定されることを特徴とする請求項1又は2に記載のデータ伝送方法。

【請求項4】 該ネットワークのコンピュータは互いに結合(リンク)されることを特徴とする請求項1～3のいずれか一つに記載のデータ伝送方法。

【請求項5】 数人の顧客(C1、C2、C3、C4、C5、C6、C7、C8、C9、C10)がリンクの少なくとも一部分を、特に有線及び無線接続の少なくとも一つを共有することを特徴とする請求項4に記載のデータ伝送方法。

【請求項6】 データ伝送の目的でリンクは最大帯域幅を有し、顧客には割り当てられた最大帯域幅(CIR)が提供されていることを特徴とする請求項4又は5に記載のデータ伝送方法。

【請求項7】 前記顧客(C1、C2、C3、C4、C5、C6、C7、C8、C9、C10)のトラフィックが測定されることを特徴とする請求項6に記載のデータ

$$p = k1 \times (b_{ref} - b) - k2 \times (b_{ref} - b_{old}) + p_{old}$$

次のステップにおいて $p_{old}$ は $p$ に等しくセットされ $b_{old}$ は $b$ に等しくセットされることを特徴とする請求項1～4のいずれか一つに記載のデータ伝送方法。

【請求項16】 確率(p)は、0と1との間に一様に分布された乱数(u)と比較されることを特徴とする請求項1～15のいずれか一つに記載のデータ伝送方法。

【請求項17】 確率(p)が乱数(u)より大きい場合に、前記パケットは前記高廃棄優先順位(アウト)で印が付けられることを特徴とする請求項16に記載のデータ伝送方法。

【請求項18】 前記パケットは、好ましくはコア・ノードに割り当てられたバッファーに入れられることを特徴とする請求項1～17のいずれか一つに記載のデータ伝送方法。

【請求項19】 伝送中におけるパケット転換の場合

伝送方法。

【請求項8】 伝送中に、前記顧客(C1、C2、C3、C4、C5、C6、C7、C8、C9、C10)に割り当てられた最大帯域幅(CIR)を越えたとき、もしくは接続の最大帯域幅がパケットを伝送するに充分ないときのいずれか少なくとも一方のとき、前記パケットが落とされることを特徴とする請求項6又は7に記載のデータ伝送方法。

【請求項9】 パケットの印付けは現在の帯域幅と割り当てられた最大帯域幅(CIR)との比較に基づくことを特徴とする請求項6～8のいずれか一つに記載のデータ伝送方法。

【請求項10】 現在の帯域幅と割り当てられた最大帯域幅(CIR)との比較はトークン・パケットに基づくことを特徴とする請求項9に記載のデータ伝送方法。

【請求項11】 現在の帯域幅が割り当てられた最大帯域幅(CIR)より高い場合に、前記パケットに前記高廃棄優先順位で印が付けられることを特徴とする請求項6～10のいずれか一つに記載のデータ伝送方法。

【請求項12】 データの送信はTCP輸送プロトコルを介して、特にTCP/IPを介して、行われることを特徴とする請求項1～11のいずれか一つに記載のデータ伝送方法。

【請求項13】 前記パケットはDiffServ環境で、好ましくはPHBを介して、特にWREDによる保証付き転送(AF)で、転送されることを特徴とする請求項1～12のいずれか一つに記載のデータ伝送方法。

【請求項14】 前記印付けは2つの状態により行われることを特徴とする請求項1～13のいずれか一つに記載のデータ伝送方法。

【請求項15】 与えられたステップでの確率(p)は、以下の数1に示される式のように表現され、  
【数1】

に、前記高廃棄優先順位(アウト)で印が付けられているパケットは捨てられることを特徴とする請求項1～18のいずれか一つに記載のデータ伝送方法。

【請求項20】 TCPウィンドウのサイズ(【外1】)

$\dot{W}$

)の変化は、以下の数2に示される式のように表現されることを特徴とする請求項1～19のいずれか一つに記載のデータ伝送方法。

【数2】

$$\dot{W} = \frac{1}{R(t)} - \frac{W(t) \cdot W(t-R(t))}{2 \cdot R(t-R(t))} p(t-R(t)).$$

【請求項21】 トークン・パケット・オキュパンシー

(  
【外2】

b

) の変化は、以下の数3に示される数式のように表現されることを特徴とする請求項1～20のいずれか一つに記載のデータ伝送方法。

【数3】

$$\dot{b} = -\frac{W(t)}{R(t)} N(t) + C.$$

$$\begin{aligned}\delta \dot{W} &= -\frac{N}{R_0^2 C} (\delta W + \delta W(t-R_0)) - \frac{R_0 C^2}{2N^2} \delta p(t-R_0), \\ \delta \dot{b} &= -\frac{N}{R_0} \delta W,\end{aligned}$$

【数5】

$$\delta W = W - W_0$$

$$\delta b = b - b_0$$

$$\delta p = p - p_0.$$

【請求項23】 以下の数6に示されるように  $1/R_0$  が  $N/R_0^2 C$  よりも著しく大きいと仮定して、

【数6】

$$\frac{N}{R_0^2 C} \ll \frac{1}{R_0},$$

伝達関数は以下の数7に示される数式のように表すことができることを特徴とする請求項1～22のいずれか一つに記載のデータ伝送方法。

【数7】

$$H(s) = -\frac{R_0 C^2}{2N^2} \frac{1}{s + \frac{2N}{R_0^2 C}} e^{-sR_0}.$$

【請求項24】 トーカン・パケット・オキュパンシー (b) は、コントローラ、特に、以下の数8に示された数式を満たすPIコントローラによって、

【数8】

$$C(s) = K \frac{\frac{s}{T} + 1}{s}.$$

安定させられることを特徴とする請求項1～23のいずれか一つに記載のデータ伝送方法。

【請求項25】 制御システム定数はTCP時定数より大きな値にセットされ、特にコントローラのゼロ点は、コントローラに閉ループの挙動を支配させるために、以下の数9に示される数式を満たすようにセットされることを特徴とする請求項24に記載のデータ伝送方法。

【数9】

【請求項22】 TCPウィンドウ・サイズ (W) 及びトーカン・パケット・オキュパンシー (b) の少なくとも一つの変化は、以下の数4及び数5に示される数式で決定される操作点で、好ましくは一定の往復遅延時間 (RTT) 及びTCPソースの一定数 (N) の少なくとも一つで線形化されることを特徴とする請求項20又は21に記載のデータ伝送方法。

【数4】

$$z = \omega_g = 0.1 \frac{2N^-}{R^+ C}$$

【請求項26】 特にナイキスト判定法を用いることによりコントローラにおける利得 (K) は以下の数10に示されるような数式を満たすようにセットされることを特徴とする請求項24又は25に記載のデータ伝送方法。

【数10】

$$K = 0.007 \frac{(2N^-)^3}{(2R^+ C^2)^2}.$$

【請求項27】 k1は、好ましくは以下の数11に示される数式を満たすような双線形変換 (bilinear transformation) により、

【数11】

$$k1 = K \left( \frac{T}{2} + \frac{1}{\omega_g} \right).$$

計算されることを特徴とする請求項15、25、又は26に記載のデータ伝送方法。

【請求項28】 k2は、好ましくは以下の数12に示される数式を満たすような双線形変換 (bilinear transformation) により計算されることを特徴とする請求項15、25、26、又は27に記載のデータ伝送方法。

【数12】

$$k2 = -K \left( \frac{T}{2} - \frac{1}{\omega_g} \right).$$

【発明の詳細な説明】

【0001】本発明は、コンピュータ・ネットワークを介して、特に送られるべきデータがIPパケットに分割されるインターネットを介して、顧客からのデータを伝送する方法に関する。更に、各パケットには少なくとも

2つの状態（イン及びアウト）により印が付けられ、その状態は、もし伝送中にネットワーク輻轆に起因してパケットが落とされる（ドロップされる）場合、どのパケットが最初に落とされるかを決定する。

【0002】今日、コンピュータ・ネットワークを介して顧客からのデータを伝送する多様な方法がある。インターネットを介して伝送されるべきデータは一般にパケットに分割される。データは、IPプロトコルを介して伝送されるならば、インターネット・プロトコル（IP）パケットに分割される。ネットワークを介してパケットがスムーズに、即ち輻轆無しで、伝送されることを保証するために、パケットは少なくとも2つの状態のうちの1つによって印が付けられる。これらの状態の目的は、伝送中にパケットが落とされるとすれば、どのパケットが最初に落とされどのパケットが最後に落とされるかを決定することである。パケットの廃棄は、ネットワーク輻轆に起因して生じる。この場合、高い廃棄優先順位が印されているパケット（アウト・パケット）は最初に捨てられ、低い廃棄優先順位が印されているパケット（イン・パケット）は捨てられないという高い確率を有する。

【0003】パケットは、ネットワークに入るときに、即ち例えばインターネット・サービス・プロバイダ（ISP）の端のノードで、印が付けられる。パケットには、例えばパケットのサイズが特定のバイト数より小さいか否かなど、それぞれのパケットが特定の条件に準拠するか否か調べるアルゴリズムに従って、印が付けられる。この条件に準拠しないパケットには、パケットがネットワーク輻轆の場合に最初に落とされる（アウト・パケット）状態で印が付けられる。

【0004】パケットに印を付ける前記のシステムは、パケットが条件を満たさないならば高い廃棄優先順位の状態で印を付けるだけであるので、特に問題である。普通そうであるように、パケットが伝送中に割り当てられている最大帯域幅を上回れば高い廃棄優先順位で印が付けられるという条件をアルゴリズムが含んでいる場合には、特にそうである。このことは、顧客の割り当てられている最大帯域幅を既に上回っているときには高い廃棄優先順位で印が付けられているパケットが落とされるということを意味する。パケットが準拠しないときにだけ高い廃棄優先順位でパケットに印を付けることは、準拠しないパケットを輻轆の場合に捨てるこことを可能にするが、輻轆を防ぐことはできない。

【0005】本発明の目的は、端のルータにおいてパケットに印を付ける方法を最適化することによって、ネットワーク輻轆を回避することを目的とする、冒頭で述べた種類のデータ伝送方法を提供することである。

【0006】本発明によれば、この目的は、高廃棄優先順位の状態でのパケットの印付けがランダム確率に基づくことを特徴とする請求項1の特徴を示すデータ伝送方

法により達成される。

【0007】本発明によれば、ランダム確率に基づいてパケットに印を付けることにより、パケットは割り当てられている最大帯域幅を上回っていないときに既に高廃棄優先順位の状態で印を付けられることができる。従って、パケットは、パケットの伝送中に割り当てられている最大帯域幅を上回っていないときに早期に落とされることもあり得る。パケットがTCPプロトコルによって転送されるならば、早期のパケット廃棄はソースに伝送速度（パケットがネットワークに送り込まれる速度）を低下させ、これはネットワーク輻轆を予防することを可能にする。これは、顧客がデータを送る帯域幅または顧客の束ねたトラフィックを制御し最適化する非常に有利で単純な方法を提供する。

【0008】特に有効な伝送を保証することに関して、高廃棄優先順位でのパケットの印付けは、各顧客について単一のランダム確率に基づき、それにより計算量を最小限にする。

【0009】ネットワークのノードはリンクにより互いに接続される。複数の顧客が特に有線及び／又は無線接続のリンクを共有することができる。この場合、各顧客について、顧客が送るトラフィックを特徴づけるために1つのランダム確率が使用される。本発明による方法は、リンクでの総帯域幅を最大にしようと試みる既存の方法とは異なって、各顧客が使用する帯域幅を最適化しようとする。換言すると、従来の方法では、他の顧客を犠牲にして一人または複数の顧客が、自分が支払った分よりも著しく大きい帯域幅を受け取るということがあり得る。本発明による方法では、各顧客は自分が支払った分の値に近い帯域幅を使用することができる。

【0010】リンクは最大帯域幅を有し、及び／又は顧客にはデータ伝送のために最大の帯域幅が割り当てられる。割り当てられた最大帯域幅に基づく課金及び支払いは特に簡単であるので、ISP及び顧客に関してはこの様なシナリオが良くある。

【0011】パケットがネットワークに入るとき、そのパケットが準拠するか否か判定するために、即ち顧客により使用される帯域幅がその顧客が支払った分の値を上回るか否か判定するために、一つの方法が適用される。一定の契約へのパケットの準拠を査定するために使用される方法はポリサーと呼ばれる。もっとも一般的なポリサーは、トークン・パケットである。パケットがネットワークに入るとき、与えられたトークン・パケット・サイズによって特徴付けられるトークン・パケットは、顧客が購入した帯域幅に対応する率で満たされる。トークン・パケットのサイズ（深さとも呼ばれる）と割り当てられた帯域幅とは、両方とも、一般にISPと顧客との間の契約の一部分である。

【0012】顧客からの与えられたサイズまたは長さのパケットが端のノードで受信されるとき、パケット長さ

(バイト単位で測られる)がトークン・パケットのバイト数、即ちトークン・パケット・オキュパンシー、を上回っていれば高廃棄優先順位でそれに印が付けられる。もしこのパケットのためにパケット内に充分なバイトがあれば、それには低廃棄優先順位で印が付けられる。もしパケットに低廃棄優先順位で印が付けられれば、パケット長さに等しい数のバイトがトークン・パケットから差し引かれる。もし高廃棄優先順位でパケットに印が付けられれば、トークン・パケットからバイトは差し引かれない。トークン・パケットが空ならば、全てのパケットが高廃棄状態で印を受けられる。

【0013】コア・ノードにおいて、全てのパケットがその印(イン/アウト)に関わらず同じバッファーに入れられる。このバッファーは、輻輳が生じた場合にアウト・パケットが最初に落とされるように管理される。この様にして、イン・パケットだけでは輻輳を生じさせないようにネットワークが構成されている限りはイン・パケットが決して落とされないことが保証される。

【0014】提案されている本発明は、標準的トークン・パケットを次のように拡張する。パケットは、端のルータに到着すると、トークン・パケットに入る。もしパケットのサイズがトークン・パケット内のバイト数を上回っていなくても、パケットは(標準的トークン・パケットの場合とは異なって)一定の確率で準拠していないと印される(高廃棄優先順位)。ネットワークの輻輳の場合には、このパケットはおそらく落とされるであろう(以下、この廃棄を早期廃棄と称する)。本発明は、もしパケットが伝送制御プロトコル(TCP)(特に伝送制御プロトコル/インターネット・プロトコル、即ちTCP/IP)により伝送されるならば、早期廃棄は重大なレベルの輻輳が発生する前にソースが伝送を抑えることを可能にするという事実に基づいている。換言すると、早期廃棄は多数のパケットが落とされる事態を防止することになる。

【0015】非常に簡単な実施態様では、パケットはディフサーブ(DiffServ)環境において転送されることができる。ディフサーブ(ディフサーブアーキテクチャとも称される。)はインターネットでサービス品質(QoS)を提供する(小規模から大規模まで)拡張可能な方法である。拡張可能性は、端の方向へ複雑な機能性を移動させると共に非常に簡単な機能性をコアに残していくことによって達成される。DiffServでは、パケットにネットワークの入り口でDiffServコードポイント(DSCP)で印が付けられ、コアにおいてはこれらのDSCPに応じてそれらに転送処理が与えられる。各DSCPはパー・ホップ挙動(Per-Hop Behavior(PHB))に対応する。

【0016】2グループのPHB、すなわち優先転送(EF)PHB及び保証付き転送(AF)PHBがこれまでに定義されている。

【0017】ディフサーブサービスを提供するサービス・プロバイダ、特にISPは、一般に保証付き転送(AF)を用いてサービスを提供する。AFでは、顧客のパケットは、該顧客からの総トラフィックが、契約された帯域幅、即ち割り当てられた最大帯域幅を上回らない限り非常に高い確率で転送される。ネットワーク輻輳の場合に、もし総トラフィックが割り当てられている最大帯域幅を上回ると、顧客の準拠しないパケットは高い確率で捨てられる。

【0018】一般に、サービス・プロバイダによるデータ伝送についての課金は契約され割り当てられた最大帯域幅に基づくので、顧客は割り当てられた最大帯域幅に少なくとも等しい伝送速度を受けると期待する。実際にには、TCPと共にAFを使用すると、平均総トラフィックが割り当てられた最大帯域幅より著しく低いという結果をもたらす。それは、パケットが落とされるときにTCPがそのトラフィックを減少させるからである。従って、TCPとAFとを組み合わせると、もし割り当てられた最大帯域幅を上回れば常に前述の挙動が行われる結果となる。場合によっては、この組み合わせは、全ての顧客のTCPソースの、全てがその送信速度を同時に低下させるという同時(同期)挙動という結果をももたらす。その結果として、顧客の送信速度が変動して、トラフィックが契約された帯域幅より著しく低くなるという結果をもたらす。

【0019】DiffServでは、前述したAFと組み合わされたTCP伝送の挙動は非常に頻繁に観察される。送信速度がCIR(Committed Information Rate(委託情報速度)、他には割り当てられた最大速度とも称される)を上回ると、トークン・パケットは空になり幾つかのパケットはアウトと印される。従って、この印付けは、割り当てられた最大帯域幅を上回るとパケット廃棄という結果をもたらす。

【0020】この印付けアルゴリズムは、3レベルの廃棄優先順位に拡張することができる。その様な解決策は、特に高度にパケットを区別することを可能にする。廃棄優先順位のレベルは如何なる数にも拡張されることができる。

【0021】コア・ノードでは、全てのパケットが、その印に関わらず、同じバッファーに入れられる。このバッファーは、輻輳が生じた場合に高廃棄優先順位の印を付されたパケットが最初に落とされるように管理される。高廃棄優先順位パケットが最初に落とされるようにバッファーを管理するために通常使用される1つのメカニズムはWRED(Weighted Random Early Detection(重み付きランダム早期検出))である。WREDは、低廃棄優先順位で印を付けられたパケットだけではパケット輻輳を生じさせないようにネットワークが構成されている限りは、これらのパケットが決して落とされないということを保証する。

【0022】TCPは、これらの廃棄に反応して、トラフィックを、割り当てられた最大帯域幅より低い値まで減少させる。TCPは、それ以上のパケット廃棄を検出しなければ、次のパケット廃棄が発生するまでトラフィックを再び増大させる。その結果として、TCPの送信速度は割り当てられた最大帯域幅と時には相当低い値との間で変動し、その結果として平均トラフィックは割り当てられた最大帯域幅より低くなる。この挙動は、付加

$$p = k_1 \times (b_{ref} - b) -$$

ここで  $p_{old}$  及び  $b_{old}$  は前のステップ（前の更新時）にそれぞれ  $p$  及び  $b$  が持った値である。次のステップを評価するために、 $p_{old}$  は  $p$  に等しくセットされなければならず、 $b_{old}$  は  $b$  に等しくセットされなければならない。 $b_{ref}$  は、望まれるトークン・パケット・オキュパンシー、即ちトラフィックを安定させるためにその値に調整したい制御ループの値である。

【0025】パケットがトークン・パケットに入るたび毎に、この確率は0と1との間に一様に分布された乱数と比較される。もし該確率がその乱数より大きければ、そのパケットは高廃棄優先順位で印を付けられる。

【0026】トークン・パケット・オキュパンシーを安定させると、TCPウィンドウのサイズの変化は、以下の数14に示された式で表すことができる。

【0027】

【数14】

$$\dot{W} := \frac{1}{R(t)} - \frac{W(t) \cdot W(t-R(t))}{2 \cdot R(t-R(t))} p(t-R(t)).$$

トークン・パケット・オキュパンシーの値の変化は、以下の数15に示された式で表すことができる。

【0028】

【数15】

$$\dot{b} = -\frac{W(t)}{R(t)} N(t) + C,$$

ここで  $W(t)$  はTCPウィンドウのサイズであり、 $R(T)$  は往復遅延時間 (RTT) であり、 $N(t)$  は顧客のTCPソースの数であり、 $C$  は割り当てられた最大帯域幅（他の場所ではCIRとも称される）である。

【0029】トークン・パケット・オキュパンシーを安定させるために、一定の往復遅延時間  $R_0$  及び／又は一定のTCPソース数  $N$  を仮定して、操作点におけるTCPウィンドウ・サイズ及び／又はトークン・パケット・オキュパンシーの値の変化を以下の数16、数17に示される式で線形化することができる。

【0030】

【数16】

的ランダム確率に基づいてパケットに印を付けることによって相当減少させられる。

【0023】総トラフィックを最適化するために、与えられたとき（ステップ）におけるランダム確率は、以下の数13に示される式で表すことができる。

【0024】

【数13】

$$k_2 \times (b_{ref} - b_{old}) + p_{old}$$

$$\delta \dot{W} = -\frac{N}{R_0^2 C} (\delta W + \delta W(t-R_0)) - \frac{R_0 C^2}{2N^2} \delta p(t-R_0),$$

$$\delta \dot{b} = -\frac{N}{R_0} \delta W,$$

【数17】

$$\delta W = \bar{W} - W_0$$

$$\delta b = b - b_0$$

$$\delta p = p - p_0.$$

操作点 ( $w_0$ 、 $b_0$ 、 $p_0$ ) は、以下の数18に示される式を満たすような条件を課すことによって決定される。TCPソースの数については  $N(t) = N$ 、往復遅延時間については  $R(t) = R_0$ 、即ちそれらは定数であると仮定する。

【0031】

【数18】

$$\dot{W} = 0, \dot{b} = 0$$

以下の数19に示される条件を仮定すると、制御ループの伝達関数は、以下の数20に示される式で表すことができる。この伝達関数は、上記の微分方程式に対してラプラス変換を実行することによって得られる。

【0032】

【数19】

$$\frac{N}{R_0^2 C} \ll \frac{1}{R_0},$$

【数20】

$$H(s) = -\frac{R_0 C^2}{2N^2} \frac{1}{s + \frac{2N}{R_0^2 C}} e^{-sR_0}.$$

非常に有利な実施態様では、トークン・パケット・オキュパンシーはコントローラにより、特に以下の数21に示される式を満たすようなPIコントローラにより安定化されることができる。ラプラス変換を実行することにより得られた  $C(s)$  の値を伴うPIコントローラは、最大入力過渡と高いsettlingタイムとを有するが、オフセットは有さない。従って、PIコントローラはトークン・パケット・オキュパンシーを安定させるの

に良く適している。

【0033】

【数21】

$$C(s) = K \frac{\frac{s}{z} + 1}{s}$$

開ループの伝達関数は以下の数22に示される式のように表される。

【0034】

【数22】

$$L(j\omega) = e^{-j\omega R_0} \frac{C^2 K \frac{j\omega}{z} + 1}{2N} \frac{1}{j\omega + \frac{2N}{R_0^2 C}}$$

TCPソースの数について  $N \geq N^-$ 、往復遅延時間 (RTT) について  $R_0 \leq R^+$  という範囲を仮定すると、目的は線形制御ループを安定させる定数K及びzの値を選択することである。

【0035】この目的のために、TCP時定数より大きい制御システム定数を選択することができ、該コントローラについてのゼロ点は、以下の数23に示される式を満たすように選択され得る。

【0036】

【数23】

$$z = \omega_g = 0.1 \frac{2N^-}{R^+ C}$$

上記選択の理論的根拠はコントローラに閉ループ挙動を支配させることであり、ここで制御定数は  $1/\omega_g$  にニアリーアイコールであり、TCP時定数は以下の数24に示される値として定義される。

【0037】

【数24】

$$\frac{2N^-}{R^+ C}$$

ナイキスト安定判別法を実施すると、システムは以下の数25に示されるK値に対する  $\omega_g$  で安定する。

【0038】

【数25】

$$K = 0.007 \frac{(2N^-)^3}{(2R^+ C^2)^2}$$

ナイキスト安定判別による基準は、  $\omega_g$  についてシステムが安定である時を定義する。方程式  $|L(j\omega_g)| = 0.1$  を課すことにより、Kについての値を得る。

【0039】位相差についての方程式を計算することにより、以下の数26に示される式が得られる。

【0040】

【数26】

$$\angle L(j\omega_g) \geq -146^\circ > -180^\circ.$$

従って、ループはこれらの値について安定である。

【0041】ラプラス領域からz領域への変換、好ましくは双一次変換、を実行することにより、以下の数27に示される式を満たすk1値及びk2値が得られる。

【0042】

【数27】

$$k1 = K \left( \frac{T}{2} + \frac{1}{\omega_g} \right), k2 = -K \left( \frac{T}{2} - \frac{1}{\omega_g} \right),$$

ここでKはコントローラにおける利得であり、  $\omega_g$  はシステムの周波数領域である。Tは、例えば到着間時間として定義されるサンプリング時間であり、これは逆最大帯域幅  $1/CIR$  に等しい、即ち顧客は自身が契約した最大帯域幅で送信する。

【0043】本発明を応用し更に発展させるいろいろな有利な方法がある。この目的のために、本発明の請求項と、図面を参照しての本発明に係る方法の好ましい実施態様についての記述を参照のこと。図面を参照しての好ましい実施態様についての記述は、該教示の一般的に好ましい実施態様も含む。

【0044】図1は、パケットがAF及びWREDでPHBを介して送られるDiffServ環境で二人の顧客C1及びC2がISPを介して顧客D1及びD2にデータを送るシナリオ例を示している。C1及びC2は、それらのISPと10Mbpsの割り当てられた最大帯域幅(CIR)を協定している。更に、顧客C1及びC2は、20ms(C1)及び100ms(C2)のRTTで、20Mbpsの最大帯域幅のリンクを共有して、共に20TCPフローを各々送信する。

【0045】シミュレーション結果によると、この模範的実施態様では、顧客C1及びC2のトラフィックはランダム確率に基づく付加的な印付け方式無しで既知のトーカン・パケット・アルゴリズムを使用するとき、各々9.83Mbps及び8.32Mbpsである。顧客C2のトラフィックは10Mbpsの割り当てられた最大帯域幅CIRより相当低いことに注意されたい。

【0046】図2は、既知のトーカン・パケット・アルゴリズムの概略図を示している。このアルゴリズムにより、実際の帯域幅が割り当てられた最大帯域幅、CIRと比較される。パケットがISPのネットワークに入るとき、サイズBのトーカン・パケットが割り当てられた最大帯域幅CIRにより明示される速度で満たされる。トーカン・パケット・サイズB及び割り当てられた最大帯域幅CIRは共にISPと顧客C1及びC2との間にそれぞれ契約されたもの的一部分である。

【0047】長さLのパケットがトーカン・パケットに入るととき、それは、もしトーカン・パケット・オキュパ

ンシーアが必要なバイトより少ないバイトを有するならば、アウトと印される（即ち、高廃棄優先順位で印が付けられる）。もしこのパケットのために充分なバイトがあれば、それにはインという印が付けられる、即ち低廃棄優先順位で印が付けられる。イン印付けの場合、パケット長さしに等しい数のバイトがトーケン・パケット・オキュパンシーbから差し引かれる。トーケン・パケット・オキュパンシーbが充分なバイトを持っていないためにアウトという印がパケットに付けられたならば、トーケン・パケット・オキュパンシーbから何も差し引かれない。

【0048】図3はパケットが図2に示されているトーケン・パケット・オキュパンシー・アルゴリズムのみに基づいて印が付けられる場合、顧客C2についてのトーケン・パケット・オキュパンシーbを描いている。該プロットは束ねられたTCPトラフィックの振動する挙動を示している。トーケン・パケットが空になると、それはTCPトラフィックがその速度を割り当てられた最大帯域幅CIRより高めているからである。輻輳の場合、高廃棄優先順位で印が付けられているパケット（アウト・パケット）は落とされる。図3は、TCPがその廃棄に反応してその速度を相当低下させることを明らかに示している。この時点では、トーケン・パケットは再びフィルアップを開始する、即ちトーケン・パケット・オキュパンシーbが増大する。TCPがその速度を再びCIRより高くなるとトーケン・パケット・オキュパンシーbが再び減少する。パケットが満杯である間、顧客C2は割り当てられた最大帯域幅CIRより低い速度で送信する。

【0049】図4において、流れ図は発明された方法によるパケットの印付けを示している。パケットがネットワークに入るとき、図2のトーケン・パケット・アルゴリズムは始めに該パケットが割り当てられた最大帯域幅CIR以内かどうか調べる。この目的のために、パケットの長さLはトーケン・パケット・オキュパンシーbと比較される（ステップS101）。もしパケットの長さLの値がトーケン・パケット・オキュパンシーbの値よりも大きければ（ステップS101でno）、該パケットにはアウトという印が付けられる。もしトーケン・パケット・オキュパンシーbが充分なバイトを持っていれば（ステップS101でyes）、該パケットがインと印されるかアウトと印されるかをランダム確率pが決定する。もし確率pが0と1との間に一様に分布する乱数uより大きければ（ステップS102でyes）、該パケットにはアウトという印が付けられ；さもなくば（ステップS102でno）、それにインという印が付けられる。もしトーケン・パケットが空ならば、ランダム確率pに関わらず全てのパケットにアウトという印が付けられる。

【0050】トーケン・パケット・オキュパンシーbを

安定させるという問題は、ランダム確率pに基づく付加的な印付け方式により達成されることができる。トーケン・パケット・オキュパンシーbを安定させるという問題は、以下の数28に示すように、トーケン・パケット・オキュパンシーの時間微分

【外3】

b

を0に等しくすることとして表現される。

【0051】

【数28】

$$b = CIR - r(t) = 0,$$

ここでトーケン・パケット・オキュパンシーbは0より大きくてトーケン・パケット・サイズBより小さく、bはトーケン・パケット・オキュパンシーであり、Bはパケット・サイズであり、r(t)は顧客の送信速度であり、CIRは顧客の契約された最大帯域幅である。

【0052】バッファー・オキュパンシーを安定させる問題（キューイング・システムにおける）はアクティブ・キュー・マネージメント（AQM）の環境下において広く研究されている。上記のトーケン・パケット・オキュパンシーbを安定させる問題は、速度r(t)で満たされるサイズB及びキャパシティーC（CIRに等しい）のキューのオキュパンシーqを安定させるという問題に変換することができる。一定の伝送遅延と、全てのアウト・パケットが落とされるということとを仮定すると、この2つの問題は实际上同等であり、それは変数の変更（q=B-b）で容易に分かる。

【0053】これらの方程式は細部では異なるけれども、アーキテクチャのレベルでは同様である。それらは、バッファー・オキュパンシーの増減を監視して、このデータを、入ってくるパケットについて廃棄確率を得るためにアルゴリズムで処理する。いろいろなAQM方式は、基本的には、廃棄確率を得るために使用されるアルゴリズムにおいて異なる。

【0054】入ってくる全てのパケットについて確率pは以下の数29に示される式にて計算される。

【0055】

【数29】

$$p = k1 \times (b_{ref} - b) - k2 \times (b_{ref} - b_{old}) + p_{old}$$

ここでp<sub>old</sub>及びb<sub>old</sub>はそれぞれp及びbが前のステップ（前の更新時）に持った値である。次のステップを評価するために、p<sub>old</sub>はpに、b<sub>old</sub>はbに等しくセットされなければならない。b<sub>ref</sub>は、それに調整したい所望のトーケン・パケット・オキュパンシーである。アウトという印を付けるとき、トーケン・パケットから何も差し引かれないということに注意されたい。

【0056】トーケン・パケット・オキュパンシーbの安定性はパラメータk1及びk2による。従って、k1

及びk<sub>2</sub>の選択は、性能目的を達成するための鍵である。図5はトークン・パケット・オキュパンシーbを安定させる線形化された制御ループのブロック図を示しており、これに基づいてk<sub>1</sub>及びk<sub>2</sub>は既に説明されたアルゴリズムに従って計算される。

【0057】図6は、発明された方法に従って、即ちランダム確率pを使用することによって、データが伝送される場合における顧客C<sub>2</sub>のトークン・パケット・オキュパンシーbを描いている。トークン・パケット・オキュパンシーは約b<sub>r e f</sub>=0.75Bの値で安定する。この場合に顧客C<sub>2</sub>により得られる総トラフィックは9.65Mbpsであり、これは、第1のシミュレーションで得られる8.32Mbpsより遙かに割り当てられた最大帯域幅に近い。図6では、図3と比べて、トークン・パケットが満杯になっている時間間隔は相当短いことに注意されたい。

【0058】割り当てられた最大帯域幅CIRになるべく近いスループットを提供する目的は、トークン・パケット・オキュパンシーbを基準値b<sub>r e f</sub>のあたりに安定させることであると再公式化されることができる。この特定の代表的実施態様ではb<sub>r e f</sub>は下記の数30に示されるとおりである。

【0059】

【数30】

$$b_{\text{ref}} \approx 0.75B$$

一定の最大限でないトークン・パケット・オキュパンシーbは、割り当てられた最大帯域幅CIRにほぼ等しいイン・パケットの送信速度を暗に意味する。イン・パケットが落とされることは殆どなさうなので、これは割り当てられた最大帯域幅CIRにほぼ等しいスループットに通じる。

【0060】従って本発明の方法は、ランダム確率pに基づくアウト印付けを介して来るべき輻輳をTCPソースに早期に知らせることに依拠する。この様にして、本発明の方法はC<sub>1</sub>及びC<sub>2</sub>それぞれのTCPソース間の同期化を回避し、その結果として、契約されたスループットの利用を良好にすると共に、顧客C<sub>1</sub>及びC<sub>2</sub>が異なるCIRを契約している場合に総帯域幅を良好に分布させる。その結果として、顧客C<sub>1</sub>とC<sub>2</sub>とが高度に公平となる。

【0061】本発明の方法の主な利点の一つは、単純であることである。各アクティブ接続の状態を保つ代わりに、本発明の方法は単に各トークン・パケットについて少數の付加的な固定されたパラメータ及び可変パラメータを必要とするだけである。もう一つの特別の利点は、その構成が顧客のトラフィックに関する具体的な知識を必要とはせず、TCPセッションの数についての下限と

往復遅延時間(RTT)についての上限とを必要とするだけであることである。

【0062】以下では、本発明の教示を更に説明するために幾つかのシミュレーション・シナリオとその結果とを説明する。トークン・パケットとREDキューを使用するDiffServ環境を仮定し続ける。この様なシナリオは多くのシミュレートされたシナリオにおいて同意されたCIRを提供するのに非常に効率が良いことが判明している。

【0063】しかし、実際に興味ある幾つかの場合に、その様な従来アーキテクチャはTCPフロー制御メカニズムの故に公平の問題を解決することはできない。ランダム確率を用いることにより、結果を著しく改善することができる。本シミュレーション・シナリオでは、REDメカニズムにおけるトラフィックを適合させるための廃棄スレショルド(しきい値)はイン・パケット廃棄を回避する値にセットされる。その上、アウトと印されているパケットについての最大スレショルドOUT<sub>max</sub>は10に等しい。最後に、本発明に係る方法のシミュレーションのために、システムが早期印付けにより速く反応するように、AQMメカニズムについてのその時点でのキュー長さが考慮される。シミュレーションはns-2を使って行われた。

【0064】以下において、始めに幾つかの不均一なシナリオを考察することにより得た幾つかのシミュレーション結果を示す。始めの3つのシナリオでは、完全に加入されたリンク、即ちCIRの合計がボトルネック容量に等しいことを仮定した。対照的に、第4のシナリオではリンクが部分的にのみ加入されたときの挙動を探求した。提案された印付け方式の性能をいろいろなパラメータの関数として査定して本項を終える。全てのシミュレーションはTCPリーノー(TCP Reno)を用いて行われた。

【0065】第1のシナリオは図7に描かれ、以下の表1によって記述されている。アクセス・リンクは遅延もパケット廃棄ももたらさない。無応答ユーザー・データグラム・プロトコル(UDP)トラフィックは、TCPフローと相互に作用し合うときに公平の問題を引き起こすということが知られている。従って、このシナリオで、TCPのみのフローまたはTCP及びUDPの混合トラフィックを送信する顧客同士の相互作用を調べる。UDPトラフィックをモデル化するために、各々1.5Mbpsで送信する定ビットレート(CBR)フローを考慮した。この場合UDP速度は同意されたCIRの75%になる。

【0066】

【表1】

	CIR (Mbps)	# of flows		RTT (ms)	no p (Mbps)	p (Mbps)
		TCP	UDP			
Total	-	-	-	-	37.88	39.47
C1	10	10	0	20	9.46	10.10
C2	10	10	0	100	7.99	9.05
C3	10	10	5	20	10.35	10.21
C4	10	10	5	100	10.06	10.09

表1は、この試験のために選択した幾つかのセッティングを伝えると共に、終わりの2つのコラムにおいて標準方法及び本発明の方法についてのトラフィックに関する結果をそれぞれ示している。表1は、ランダム確率pの使用は顧客がTCPフローを送信して総帯域幅のうちより大きな部分を取ることを助けることを示している。特に、小さなRTTを特徴とするC1は同意されたCIRを達成するのに対してC2は、標準方法により許される80%とは対照的に、その90%以上を得る。

【0067】第2のシナリオでは、割り当てられた最大帯域幅CIRについて不均一な値を仮定する。いろいろ

な顧客が割り当てられた最大帯域幅CIRについて雑多な値を契約するときにも公平問題が発生する。実際、低いCIR値を特徴とする顧客は、同意されたCIRを達成する上で有利である。次のシナリオはこの挙動の例である。ボトルネック・リンク速度は22Mbpsに等しくセットされる。以下の表2は、考察される場合において、本発明の方法が総リンク利用を15%以上改善することを可能にすると共に特に顕著に公平な帯域幅分配をもたらすことを示している。

【0068】

【表2】

	CIR (Mbps)	# of flows TCP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	18.18	21.62
C1	10	10	20	8.63	10.16
C2	10	10	100	7.07	9.23
C3	1	10	20	1.43	1.16
C4	1	10	100	1.03	1.06

第3のシミュレーション・シナリオでは、顧客の数の影響を調べる。顧客及びフローの数が大きな値に増えると、標準的トクン・パケットが使用されるときでも多重化利得はリンク利用及び帯域幅分配を良好にする明確な効果を有する。ボトルネック・リンク速度は100Mbpsに等しくセットされる。以下の表3において、こ

れを確認するシミュレーション結果を示す。しかし、この場合にも、本発明の方法は全体としての性能を僅かに改善する。

【0069】

【表3】

	CIR (Mbps)	# of flows TCP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	97.17	98.63
C1	10	40	20	10.36	10.58
C2	10	10	100	9.16	9.25
C3	10	10	20	9.91	10.10
C4	10	40	100	10.11	10.27
C5	10	20	20	10.20	10.33
C6	10	20	100	9.79	9.89
C7	10	15	20	10.11	10.25
C8	10	15	100	9.47	9.66
C9	10	5	20	8.88	9.05
C10	10	10	100	9.14	9.22

第4のシミュレーションでは、帯域に余裕のあるリンク(under-subscribed link)におけるTCPフローだけまたはUDPフローだけを有する顧客同士の相互作用を調べる。53Mbpsのリンク速度を考察したが、以下の数31に示される値(75%加入リンク)であった。C3及びC4は共に各々1.5Mbpsの速度で10個の

CBRフローを送信する、即ちその送信速度はCIRよりも僅かに大きい。

【0070】

【数31】

$$\sum_{i=1}^4 CIR_i = 40 \text{ Mbps}$$

以下の表4は、TCPが超過の帯域幅のうちの標準的アプローチと比べて顕著に高い部分を得ることを本発明の方法が可能にするということを示している。これは、小さなRTTを有するC1などの意欲的なTCP顧客につ

	CIR (Mbps)	# of flows TCP + UDP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	49.56	51.31
C1	10	10+0	20	11.34	14.30
C2	10	10+0	100	9.72	10.48
C3	10	0+10	20	14.25	13.44
C4	10	0+10	100	14.24	13.08

次に、本発明により与えられる利益を顧客の数の関数として調べる。この目的のために、第3のシナリオについて実施されたセッティングを考察した。C1及びC2によりそれぞれ達成されるスループットを顧客の総数の関数として査定した。顧客C1は低いRTTと多數のデータフローとを特徴とし、従って顧客は割り当てられた最大帯域幅CIRを大いに達成しそうである。逆にC2は大きなRTTと割合に少數のフローとを有し、従って顧客は帯域幅分配に関しては不利な立場に置かれる。このシミュレーションでは、常にn人の顧客を有するシナリオについて表3における始めのn人の顧客を考察した。

【0072】図8において、本発明の方法と標準的トクン・パケットとを使用するときにC1及びC2により得られるスループットを比較する。本発明の方法は常に最善の性能を達成することを可能にする。しかし、最も顕著な改善は顧客の総数が8未満であるときに顧客C2により達成される。本発明の方法を使用することにより、顧客C2は常に割り当てられた最大帯域幅CIRの少なくとも90%を得、一方標準的トクン・パケットは顧客の総数が少ないとにはそれをかなり不利にする。後者の場合はISPアクセス・リンクでは概して一般的である。

【0073】更に、第3のシミュレーションについて総リンク利用率も評価した。結果は図9で報告されている。本発明の方法による改善は顕著である。

【0074】今、数ユニット（例えば家庭ユーザーなど）の程度の、顧客あたりに少數のフローの効果を考察する。特に、比較的に少數のフローを送信する一人の顧客を除いて各々10個のフローを送信する10人の顧客を有するシナリオの性能を分析する。全ての顧客に10MbpsのCIRが割り当てられ、RTTはいろいろな顧客につき20及び100msの間でまちまちであり、ボトルネック速度リンクは100Mbpsに等しい。

【0075】少數のフローを送信する顧客について、本発明の方法を使用するとき標準的トクン・パケットと比べて、達成されるスループットをフローの数の関数と

いて特に当てはまり、一方、割合に少數のデータフローと大きなRTTと（それぞれ10及び100ms）を有するC2は割り当てられた最大帯域幅CIRを達成できるに過ぎない。

#### 【0071】

【表4】

して評価する。結果は図10で報告されている。期待通りに、フローの数が少ないとには、得られるスループットは割り当てられた最大帯域幅CIRより相当少ない。しかし、本発明の方法を使用することにより、関連のある改善を認める。このシミュレーションでは、5個のフローを送信することにより、顧客は既に割り当てられた最大帯域幅CIRを得るが、早期印付けが行われないときには達成されるスループットは依然として割り当てられた最大帯域幅CIRより10%低い。

【0076】本発明の附加的な有利な実施態様について、反復を避けるために、本解説の一般項とこの請求項とを参照されたい。

【0077】最後に、上記の模範的な実施態様は請求されている教示を説明するために役立つに過ぎず、模範的実施態様には限定されない。

【0078】尚、以下に、本詳細な説明中に使用された符号とその意味について列挙する。

b : トクン・パケット・オキュパンシー

b<sub>old</sub> : 旧トクン・パケット・オキュパンシー

b<sub>ref</sub> : トクン・パケット・オキュパンシーをそれに調整したいところの値

B : トクン・パケットのサイズ

CIR : 割り当てられた最大帯域幅

C1, C2, ..., C10 : 顧客（送信者）

D1, D2 : 顧客（受信者）

イン : 低廃棄優先順位の状態

K : コントローラにおける利得

L : パケット長さ

N : TCPソースの数

アウト : 高廃棄優先順位の状態

p : 確率

p<sub>old</sub> : 旧確率

R, RTT : 往復遅延時間

u : 均一に分布する乱数

W : TCPウィンドウのサイズ

z : コントローラのゼロ点

$\omega_g$  : 最大周波数  
 AF : 保証付き転送  
 AQM : アクティブ・キュー・マネージメント  
 CBR : 定ビットレート  
 DiffServ : ディフサーブ  
 DSCP : ディフサーブサービス・コードポイント  
 IP : インターネット・プロトコル  
 ISP : インターネット・サービス・プロバイダ  
 PHB : パーホップ挙動  
 QoS : サービス品質  
 TCP : 伝送制御プロトコル  
 WRED : 重み付きランダム早期検出

## 【図面の簡単な説明】

【図1】既知の方法と発明された方法とによるデータの送信の模範的実施態様を示した図である。

【図2】既知のトークン・パケット・アルゴリズムの大要を示した図である。

【図3】ランダム確率無しでの既知の方法による送信におけるトークン・パケット・オキュパンシーを時間の関

数として示したグラフである。

【図4】発明された方法によるパケットの印付けを示す略流れ図である。

【図5】トークン・パケット・オキュパンシーを安定させる線形化された制御ループの略ブロック図である。

【図6】発明された方法によるデータの送信におけるトークン・パケット・オキュパンシーを時間の関数として示したグラフである。

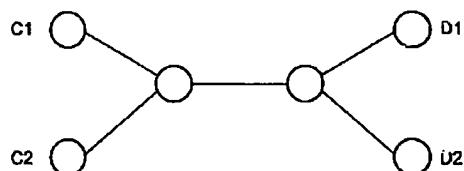
【図7】既知の方法と発明された方法とのデータ送信の付加的な模範的実施態様を示した図である。

【図8】発明された方法と比べて既知の方法を使用するときの顧客数の関数として達成されるスループットを示したグラフである。

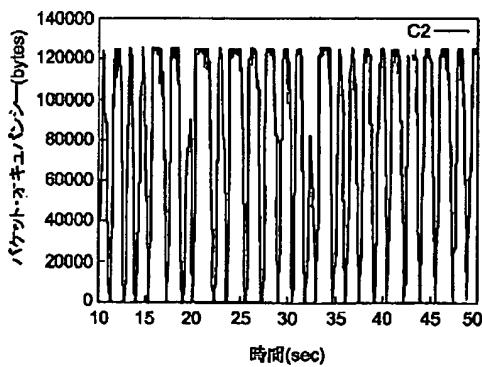
【図9】発明された方法と比べて既知の方法を使用するときの顧客数の関数として達成される総リンク利用を示したグラフである。

【図10】発明された方法と比べて既知の方法を使用するときのTCPデータフローの数の関数として達成されるスループットを示したグラフである。

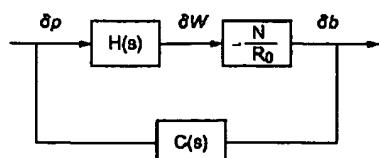
【図1】



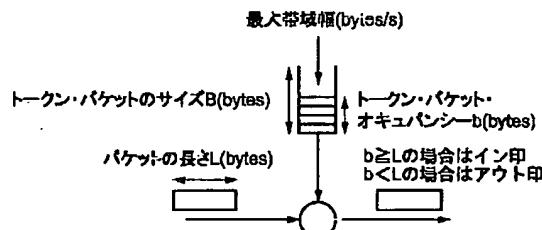
【図3】



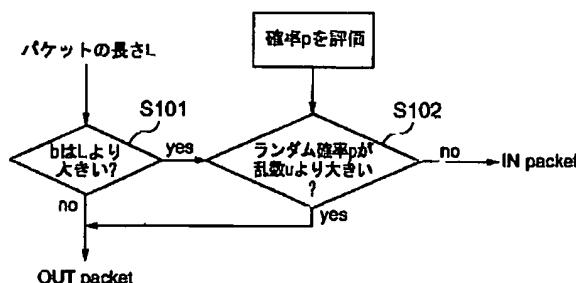
【図5】



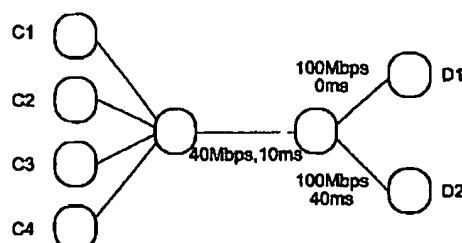
【図2】



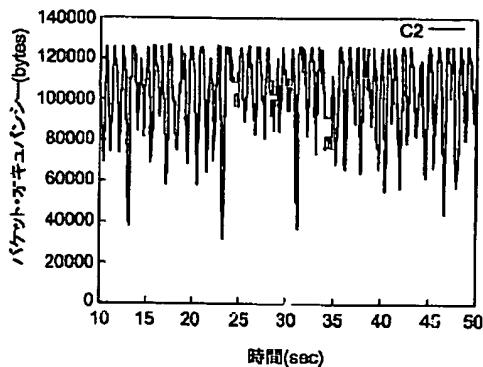
【図4】



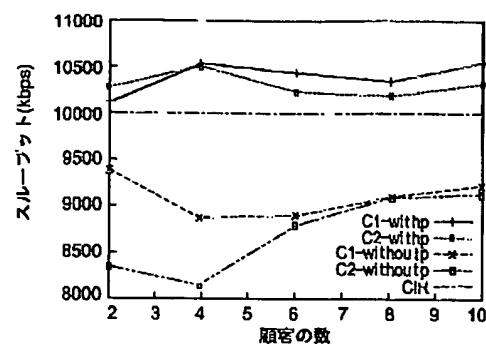
【図7】



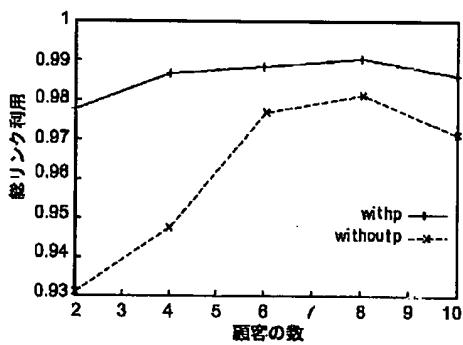
【図6】



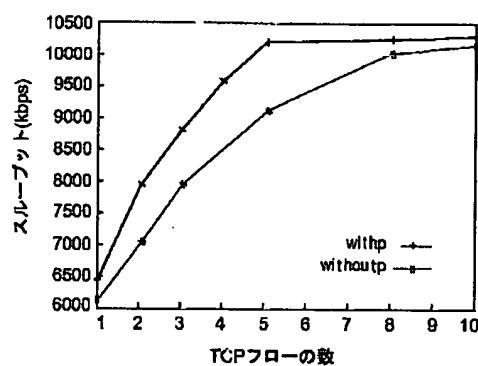
【図8】



【図9】



【図10】



フロントページの続き

(72)発明者 アルバート バンクス  
 ドイツ共和国, 69115 ハイデルベルク,  
 アデナウアー プラッツ 6, エヌイーシー  
 - ヨーロッパ リミテッド内

F ターム(参考) 5K030 GA13 HA08 HB17 JA05 KA03  
 LA03 LC03 LC13 LE17 MB02  
 MB09

【外国語明細書】

**1. Title of Invention**

METHOD OF TRANSMITTING DATA

**2. Claims**

1. Method of transmitting data from customers (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10) over a computer network, in particular over the Internet, where the data to be sent is split into packets, in particular into IP packets, where each packet is marked by one of at least two states (IN, OUT) and where the states determine which packets are dropped first, if packets are dropped during transmission, characterized in that the marking of the packet with a state (OUT) of high drop precedence is based on a random probability (p).
2. Method according to claim 1, characterized in that the marking of the packet with a high drop precedence (OUT) is based on a single random probability (p).
3. Method according to claims 1 and 2, characterized in that the probability (p) is measured for the traffic of each customer (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10).
4. Method according to claims 1 to 3, characterized in that the computers of the network are linked with each other.
5. Method according to claim 4, characterized in that several customers (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10) share at least parts of a link, in particular a line and/or a wireless connection or similar.

6. Method according to claims 4 or 5,  
characterized in that the links have maximum bandwidths  
and/or the customers have been provided an assigned maximum  
bandwidth (CIR) for the purpose of data transmission.

7. Method according to claim 6,  
characterized in that the traffic of the customer (C1, C2,  
C3, C4, C5, C6, C7, C8, C9, C10) is measured.

8. Method according to claims 6 or 7,  
characterized in that a packet is dropped when the assigned  
maximum bandwidth (CIR) of the customer (C1, C2, C3, C4, C5,  
C6, C7, C8, C9, C10) is exceeded during the transmission  
and/or the maximum bandwidth of the connection is not  
sufficient to transmit the packets.

9. Method according to claims 6 to 8,  
characterized in that the marking of the packet is based on  
the comparison of the current bandwidth with the assigned  
maximum bandwidth (CIR).

10. Method according to claim 9,  
characterized in that the comparison of the current bandwidth  
with the assigned maximum bandwidth (CIR) is based on a token  
bucket.

11. Method according to one of the claims 6 to 10,  
characterized in that the packet is marked with a high drop  
precedence if the current bandwidth is higher than the  
assigned maximum bandwidth (CIR).

12. Method according to one of the claims 1 to 11,  
characterized in that the sending of the data is done via

the TCP transport protocol and in particular via TCP/IP.

13. Method according to one of the claims 1 to 12, characterized in that the packets are forwarded in a DiffServ environment, preferably via PHB, in particular with Assured Forwarding (AF) with WRED.

14. Method according to one of the claims 1 to 13, characterized in that the marking is done by means of two states.

15. Method according to claims 1 to 14, characterized in that the probability ( $p$ ) at a given step is expressed as

$$p = k_1 \times (b_{ref} - b) - k_2 \times (b_{ref} - b_{old}) + p_{old}$$

and that at the next step  $p_{old}$  is set equal to  $p$  and  $b_{old}$  is set equal to  $b$ .

16. Method according to one of the claims 1 to 15, characterized in that the probability ( $p$ ) is compared to a random number ( $u$ ) evenly distributed between 0 and 1.

17. Method according to claim 16, characterized in that the packet is marked with a high drop precedence (OUT) if the probability ( $p$ ) is greater than the random number ( $u$ ).

18. Method according to one of the claims 1 to 17, characterized in that the packets are put into a buffer preferably assigned to the core node.

19. Method according to one of the claims 1 to 18,

characterized in that, in case of packet congestion during transmission, the packets marked with a high drop precedence (OUT) are discarded.

20. Method according to one of the claims 1 to 19, characterized in that the change in the TCP window size (W) is expressed as

$$\dot{W} = \frac{1}{R(t)} - \frac{W(t) \cdot W(t-R(t))}{2 \cdot R(t-R(t))} p(t-R(t)).$$

21. Method according to one of the claims 1 to 20, characterized in that the change in the token bucket occupancy (b) is expressed as

$$\dot{b} = -\frac{W(t)}{R(t)} N(t) + C.$$

22. Method according to claim 20 and/or claim 21, characterized in that the change in the TCP window size (W) and/or the token bucket occupancy (b) is linearized in the operation point, preferably at constant round trip time (RTT) and/or constant number of TCP sources (N)

$$\begin{aligned}\delta \dot{W} &= \frac{N}{R_0^2 C} (\delta W + \delta W(t-R_0)) - \frac{R_0 C^2}{2N^2} \delta p(t-R_0), \\ \delta \dot{b} &= -\frac{N}{R_0} \delta W,\end{aligned}$$

where

$$\delta W = W - W_0$$

$$\delta b = b - b_0$$

$$\delta p = p - p_0.$$

23. Method according to one of the claims 1 to 22,  
 characterized in that, assuming  $\frac{N}{R_0^2 C} \ll \frac{1}{R_0}$ , the transfer  
 function can be expressed as

$$H(s) = -\frac{R_0 C^2}{2N^2} \frac{1}{s + \frac{2N}{R_0^2 C}} e^{-sR_0}$$

24. Method according to one of the claims 1 to 23,  
 characterized in that the token bucket occupancy (b) is  
 stabilized by way of a controller, especially a PI controller

$$C(s) = K \frac{\frac{s}{z} + 1}{s}$$

25. Method according to claim 24,  
 characterized in that the control system constant is set to  
 a value greater than the TCP time constant and especially  
 that the zero of the controller is set to

$$z = \omega_g = 0.1 \frac{2N^-}{R^+ C^2}$$

in order to have the controller dominate the closed-loop  
 behavior.

26. Method according to claims 24 or 25,  
 characterized in that especially by invoking the Nyquist  
 criterion the gain (K) in controller is set to

$$K = 0.007 \frac{(2N^-)^3}{(2R^+ C^2)^2}$$

27. Method according to claims 15, 25 and 26,  
characterized in that k1 is computed by a preferably bilinear  
transformation as

$$k_1 = K \left( \frac{T}{2} + \frac{1}{\omega_g} \right).$$

28. Method according to claims 15, 25 and claim 26 or claim  
27,  
characterized in that k2 is computed by a preferably bilinear  
transformation as

$$k_2 = -K \left( \frac{T}{2} - \frac{1}{\omega_g} \right).$$

### **3. Detailed Description of Invention**

The invention relates to a method of transmitting data from customers over a computer network, in particular over the Internet, where the data to be sent is split into IP packets. We assume further that each packet is marked by one of at least two states (in and out) and the states determine which packets are dropped first, if packets are dropped during transmission due to network congestion.

Today, there are many different methods of transmitting data from a customer over a computer network. Data to be transmitted over the Internet is generally split into packets. If the data is transmitted via the IP protocol, it is split into Internet Protocol (IP) packets. In order to guarantee the smooth transmission of the packets over the network, i.e. without congestion, the packets can be marked by one of at least two states. The objective of these states is to determine which packets are dropped first and which packets

are dropped last, if packets are dropped during transmission. Packet drops occur due to network congestion. In this case, packets marked with a high drop precedence (out packets) are discarded first while packets marked with a low drop precedence (in packets) have a higher probability of not being discarded.

Packets are marked at their entry into the network, i.e. at the edge node of an Internet Service Provider (ISP), for example. The packets are marked according to an algorithm that checks if the respective packet conforms to a specific condition, e.g. if the size of the packet is smaller than a specific number of bytes. Packets that do not conform to this condition are marked with the state where a packet is dropped first (out packets) in case of network congestion.

The described system of marking packets is especially problematic in that packets are only marked with a state of high drop precedence if the packet does not meet the condition. This is especially true if an algorithm contains the condition, as is usually the case, that a packet is marked with a high drop precedence if the packet exceeds an assigned maximum bandwidth during transmission. This means that packets marked with a high drop precedence are dropped when the assigned maximum bandwidth of the customer has already been exceeded. Marking packets with a high drop precedence only when packets are not conforming, allows to discard non conforming packets in case of congestion, but does not allow to prevent congestion.

#### **Summary of the Invention**

The object of the present invention is to provide a method of transmitting data of the kind mentioned in the beginning

that aims at avoiding network congestion, by optimizing the way packets are marked at the edge router.

According to the present invention, this object is achieved by a data transmission method displaying the features of claim 1, characterized in that the marking of the packet with a state of high drop precedence is based on a random probability.

According to the present invention, by marking the packet on the basis of a random probability, packets can already be marked with a state of high drop precedence when the assigned maximum bandwidth is not exceeded. Consequently, packets can also be early dropped when the assigned maximum bandwidth is not exceeded during the transmission of the packet. If the packets are transported by the TCP protocol, early packet drops will cause the source to slow down the transmission rate (speed at which packets are sent into the network) and this allows to prevent network congestion. This provides a very advantageous and simple method of controlling and optimizing the bandwidth at which a customer sends data or the aggregate traffic of a customer.

With regard to guaranteeing an especially effective transmission, the marking of the packet with a high drop precedence is based on a single random probability for each customer, thereby minimizing the computational workload.

The nodes in a network are connected with each other through links. Several customers could share a link, in particular a line and/or a wireless connection or similar. In this case, one random probability for each customer is used to characterize the traffic he sends. The proposed method aims at optimizing the bandwidth experienced by each customer,

unlike existing methods that attempt to maximize the total bandwidth on the link.

In other words, with traditional methods, it is more likely that one or more customers receive significantly more bandwidth than what he/they paid for at the expense of other customers. With the proposed method each customer should experience a bandwidth close to the value he paid for.

The links have maximum bandwidths and/or the customers could be assigned a maximum bandwidth for the data transmission. Such scenarios are frequent with ISPs and their customers, as charging and paying on the basis of an assigned maximum bandwidth is especially simple.

When a packet enters a network, a method is applied to decide whether the packet is conforming or not, that is to determine whether the bandwidth used by a customer exceeds or not the value the customer has paid for. The methods used to assess the conformity of a packet to a certain contract are called policers. One of the most common policers, is the token bucket. When a packet enters the network, a token bucket, characterized by a given token bucket size, is filled at a rate, corresponding to the bandwidth purchased by a customer. Both the size (also called depth) of the token bucket and the assigned bandwidth are generally part of the contract between an ISP and a customer.

Then, when a packet of a given size or length from the customer is received at the edge node, it is marked with a high drop precedence if the packet length (measured in bytes) exceeds the number of bytes of the token bucket, i.e. the token bucket occupancy. If there are enough bytes in the bucket for this packet, it is marked with a low drop precedence. If the packet is marked with a low drop precedence, a number of bytes

equal to the packet length is subtracted from the token bucket. If a packet is marked with a high drop precedence, no bytes are subtracted from the token bucket. If the token bucket is empty, all packets are marked with a state of high drop.

At core nodes, all packets are put into the same buffer independently of their marking (*in/out*). This buffer is managed in such a way that in case of congestion, *out* packets are dropped first. In this way, it is guaranteed that as long as the network is configured such that *in* packets alone do not cause congestion, *in* packets are never dropped.

The proposed invention enhances the standard token bucket as follows. When a packet arrives at an edge router, it enters a token bucket. If the size of the packet does not exceed the number of bytes in the token bucket, the packet (unlike in the standard token bucket) might still be marked as not conforming (high drop precedence) with a certain probability. In case of network congestion this packet will most likely be dropped (we refer to this drop as an early drop). The invention is based on the fact that, if packets are transported by the Transmission Control Protocol (TCP) (in particular Transmission Control Protocol / Internet Protocol, i.e. TCP/IP), then an early drop allows the source to slow down the transmission, before a serious level of congestion occurs. In other words, an early drop should prevent a situation in which many packets are dropped.

In a very simple embodiment, the packets could be forwarded in a Differentiated Services (DiffServ) environment. DiffServ - also referred to as the DiffServ architecture - is a scalable way of providing Quality of Service (QoS) in the Internet. Scalability is achieved by moving complicated functionality toward the edge and leaving the core with very

simple functionality. With DiffServ, packets are marked at the ingress of the network with a DiffServ codepoint (DSCP) and at the core they are given a forwarding treatment according to their DSCP. Each DSCP corresponds to a Per-Hop Behavior (PHB).

Two groups of PHB have been defined so far: PHB with Expedited Forwarding (EF) and PHB with Assured Forwarding (AF).

Service providers, especially ISPs, that offer DiffServ services, generally use Assured Forwarding (AF) to provide a service. With AF, the packets of a customer are forwarded with a very high probability as long as the aggregate traffic from the customer does not exceed the contracted bandwidth, i.e. the assigned maximum bandwidth. If the aggregate traffic exceeds the assigned maximum bandwidth, in case of network congestion, non conforming packets of the customer are discarded with high probability.

In general, charging for the transmission of data by a service provider is based on the contracted and assigned maximum bandwidth, therefore a customer would expect to receive a transmission rate at least equal to the assigned maximum bandwidth. In practice, AF used with TCP results in an average aggregate traffic that is substantially lower than the assigned maximum bandwidth. This is because TCP decreases its traffic when packets are dropped. The combination of TCP and AF therefore always results in the behavior described above if the assigned maximum bandwidth is exceeded. In some situations, this combination even results in a synchronized behavior of all the customer's TCP sources, that all decrease their sending rate at the same time. As a consequence, the customer's sending rate is oscillating, which results in a substantially lower traffic than the contracted bandwidth.

In DiffServ, the behavior of TCP transmission in combination with AF described above can be observed very frequently. If the sending rate exceeds the CIR (Committed Information Rate, referred elsewhere also as assigned maximum bandwidth), the token bucket is emptied and some packets are marked as out. Consequently, this marking leads to packets drops when the assigned maximum bandwidth is exceeded.

The marking algorithm could be extended to three levels of drop precedence. Such a solution would enable an especially high degree of differentiating packets. The levels of drop precedence might even be extended to any number.

At the core node, all packets are put into the same buffer independently of their marking. This buffer is managed in such a way that in case of congestion, packets marked with a high drop precedence are discarded first. One mechanism which is typically used to manage a buffer, so that high drop precedence packets are dropped first, is WRED (Weighted Random Early Detection). WRED guarantees that, as long as the network is configured such that packets marked with a low drop precedence alone do not cause packet congestion, these packets are never dropped.

TCP reacts to these drops by decreasing the traffic to a value lower than the assigned maximum bandwidth. If TCP does not detect any further packet drops, it increases the traffic again until the next packet drops occur. As a consequence, TCP's sending rate oscillates between the assigned maximum bandwidth and a value sometimes substantially lower, resulting in an average traffic lower than the assigned maximum bandwidth. This behavior is reduced substantially by marking the packets on the basis of an additional random probability.

To optimize the aggregate traffic, the random probability at a given time (step) could be expressed as

$$p = k_1 \times (b_{ref} - b) - k_2 \times (b_{ref} - b_{old}) + p_{old}$$

where  $p_{old}$  and  $b_{old}$  are the values that respectively  $p$  and  $b$  had at the previous step (previous update time). To evaluate the next step,  $p_{old}$  has to be set equal to  $p$  and  $b_{old}$  equal to  $b$ .  $b_{ref}$  is the desired token bucket occupancy, i.e. the value of the control loop to which we want to regulate in order to stabilize the traffic.

Every time a packet enters the token bucket, this probability is compared with a random number evenly distributed between 0 and 1. If the probability is greater than the random number, the packet is marked with a high drop precedence.

When stabilizing the token bucket occupancy, the change in the size of the TCP window could be expressed as

$$\dot{W} = \frac{1}{R(t)} - \frac{W(t) \cdot W(t-R(t))}{2 \cdot R(t-R(t))} p(t-R(t)).$$

The change of value of the token bucket occupancy could be expressed as

$$\dot{b} = -\frac{W(t)}{R(t)} N(t) + C ,$$

where  $W(t)$  is the TCP window size,  $R(T)$  is the round-trip time (RTT),  $N(t)$  is the number of TCP sources of the customer and  $C$  is the assigned maximum bandwidth (also referred to as CIR elsewhere).

In order to stabilize the token bucket occupancy, we could linearize the change of value of the TCP window size and/or the token bucket occupancy at the operation point, assuming a constant round trip time  $R_0$  and/or constant number of TCP sources  $N$ ,

$$\begin{aligned}\delta \dot{W} &= \frac{N}{R_0^2 C} (\delta W + \delta W(t-R_0)) - \frac{R_0 C^2}{2N^2} \delta p(t-R_0), \\ \delta \dot{b} &= -\frac{N}{R_0} \delta W,\end{aligned}$$

where

$$\delta W = W - W_0$$

$$\delta b = b - b_0$$

$$\delta p = p - p_0.$$

The operation point  $(W_0, b_0, p_0)$  is determined by imposing the conditions  $\dot{W}=0$   $\dot{b}=0$ . For the number of TCP sources we assume  $N(t)=N$  and for the round trip time  $R(t)=R_0$ , i.e., they are constant.

Assuming that  $\frac{N}{R_0^2 C} \ll \frac{1}{R_0}$ , the transfer function of the control loop could be expressed as

$$H(s) = -\frac{R_0 C^2}{2N^2} \frac{1}{s + \frac{2N}{R_0^2 C}} e^{-sR_0}.$$

This transfer function is obtained by performing a Laplace transform on the above differential equation.

In a very advantageous embodiment, the token bucket occupancy could be stabilized by a controller, especially a PI controller

$$C(s) = K \frac{\frac{s}{z} + 1}{s}$$

A PI controller with the value of  $C(s)$  obtained by performing a Laplace transform would have a maximum input transient and a high settling time, but no offset. Therefore, the PI controller is well fitted to stabilize the token bucket occupancy.

The transfer function of the open loop is expressed as follows:

$$L(j\omega) = e^{-j\omega R_0} \frac{C^2 K}{2N} \frac{\frac{j\omega}{z} + 1}{j\omega} \frac{1}{j\omega + \frac{2N}{R_0^2 C}}$$

Assuming a range for the number of TCP sources of  $N \geq N^-$  and a round-trip time (RTT) of  $R_0 \leq R^+$ , the objective is to select values for the constants  $K$  and  $z$  to stabilize the linear control loop.

To this end, we could select a control system constant greater than the TCP time constant and the zero for the controller could be chosen

$$z = \omega_g = 0.1 \frac{2N^-}{R^{+2} C}$$

The rationale behind the above choice is to have the

controller dominate the closed-loop behavior, where the control constant is defined as  $\approx 1/\omega_g$  and the TCP time constant as  $\frac{2N^-}{R^+C}$ .

By invoking the Nyquist stability criterion, the system is stable at  $\omega_g$  for

$$K = 0.007 \frac{(2N^-)^3}{(2R^+C^2)^2}.$$

The Nyquist criterion defines when a system is stable for the highest frequency  $\omega_g$ . By imposing the equation  $|L(j\omega_g)| = 0.1$ , we obtain the value for K.

By computing the equation for the phase difference, we obtain

$$\angle L(j\omega_g) \geq -146^\circ > -180^\circ.$$

Consequently, the loop is stable for these values.

By performing a transformation from the Laplace domain into the z domain, preferably a bilinear transformation, we obtain k1 and k2 as

$$k1 = K \left( \frac{T}{2} + \frac{1}{\omega_g} \right), \quad k2 = -K \left( \frac{T}{2} - \frac{1}{\omega_g} \right),$$

where K is the gain in the controller and  $\omega_g$  is the frequency domain of the system. T is the sampling time, defined for instance as the interarrival time, which is equal to the inverse maximum bandwidth  $1/CIR$ , i.e. the customer is transmitting at his maximum contracted bandwidth.

There are different advantageous ways in which to apply and further develop the teachings of the present invention. To this end, please refer to the claims at the end of this document as well as to the description of preferred embodiments of the invented method with reference to drawings that follows. The description of preferred embodiments with reference to drawings also includes the generally preferred embodiments of the teachings.

Fig. 1 shows a simulated scenario with two customers C1 and C2 sending data over an ISP to customers D1 and D2 in a DiffServ environment where the packets are sent via PHB with AF and WRED. C1 and C2 have agreed a maximum assigned bandwidth (CIR) of 10 Mbps with their ISP. In addition, customers C1 and C2 share a link of a maximum bandwidth of 20 Mbps, both sending 20 TCP flows each, with RTTs of 20 ms (C1) and 100 msec (C2).

According to the simulation results, in this exemplary embodiment the traffic of customers C1 and C2 are 9.83 and 8.32 Mbps each when using the known token bucket algorithm without an additional marking scheme on the basis of a random probability. Note that the traffic of customer C2 is substantially lower than the assigned maximum bandwidth CIR of 10 Mbps.

Fig. 2 shows a schematic depiction of the known token bucket algorithm. By means of this algorithm, the actual bandwidth is compared with the assigned maximum bandwidth, the CIR. When a packet enters the ISP's network, a token bucket of the size ~~B~~ is filled at the rate specified by the assigned maximum bandwidth CIR. Both the token bucket size B and the assigned maximum bandwidth CIR are part of the respective

contract between the ISP and the customers C1 and C2.

Then, when a packet of the size of length  $L$  enters the token bucket, it is marked out (i.e. marked with a high drop precedence) if the token bucket occupancy  $b$  has less bytes than required. If there are enough bytes in the bucket for this packet, it is marked in, i.e. marked with a low drop precedence. In case of in marking, a number of bytes equal to the packet length  $L$  is subtracted from the token bucket occupancy  $b$ . If a packet is marked out because the token bucket occupancy  $b$  does not have enough bytes, no bytes are subtracted from the token bucket occupancy  $b$ .

Fig. 3 plots the token bucket occupancy  $b$  for customer C2 if the packets are marked solely on the basis of the token bucket occupancy algorithm shown in Fig. 2. The plot shows the oscillating behavior of the TCP traffic aggregate. When the token bucket gets empty it is because the TCP traffic has increased its rate over the assigned maximum bandwidth CIR. In case of congestion, packets marked with a high drop precedence (out packets) are dropped. Fig. 3 clearly shows that TCP reacts to the drops by significantly decreasing its rate. At this point, the token bucket starts filling up again, i.e. the token bucket occupancy  $b$  increases. It is not until TCP increases its rate over the CIR again that the token bucket occupancy  $b$  decreases again. In the time period while the bucket is full the customer C2 is transmitting at a lower rate than the assigned maximum bandwidth CIR.

In Fig. 4, a flow diagram shows the marking of a packet according to the invented method. When the packet enters the network, the token bucket algorithm of Fig. 2 first checks if the packet is within the assigned maximum bandwidth CIR. To this end, the packet length  $L$  is compared with the token

bucket occupancy  $b$ . If the value of the packet length  $L$  is greater than the value of the token bucket occupancy  $b$ , the packet is marked out. If the token bucket occupancy  $b$  has enough bytes, the random probability  $p$  determines whether the packet is marked in or out. Now, if the probability  $p$  is greater than a random number  $u$  evenly distributed between 0 and 1, the packet is marked out; otherwise it is marked in. If the token bucket is empty, all packets are marked out independently of the random probability  $p$ .

The problem of stabilizing the token bucket occupancy  $b$  can be achieved by an additional marking scheme on the basis of the random probability  $p$ . The problem of stabilizing the token bucket occupancy  $b$  can be expressed as having the time derivative of the token bucket occupancy  $b$  equal to 0:

$$\dot{b} = \text{CIR} - r(t) = 0,$$

where the token bucket occupancy  $b$  is greater than 0 and smaller than the token bucket size  $B$ , where  $b$  is the token bucket occupancy,  $B$  is the bucket size,  $r(t)$  is the sending rate of the customer and the CIR is the contracted maximum bandwidth of the customer.

The problem of stabilizing the buffer occupancy (in a queuing system) has been extensively studied in the context of Active Queue Management (AQM). The problem of stabilizing the token bucket occupancy  $b$  described above can be transformed into the problem of stabilizing the occupancy  $q$  of a queue of size  $B$  and capacity  $C$  (equal to CIR) filled at a rate  $r(t)$ . Assuming constant round trip delays and that all out packets are dropped, the two problems are actually equivalent, which can be easily seen with the change of variable:

$$q=B-b.$$

While these schemes differ in details, they are similar at the architectural level. They monitor the evolution of the buffer occupancy and process this data with an algorithm to obtain a dropping probability for incoming packets. Different AQM schemes basically differ in the algorithm used to obtain the dropping probabilities.

For every incoming packet the probability  $p$  is computed as

$$p = k_1 \times (b_{ref} - b) - k_2 \times (b_{ref} - b_{old}) + p_{old}$$

where  $p_{old}$  and  $b_{old}$  are the values that respectively  $p$  and  $b$  had at the previous step (previous update time). To evaluate the next step,  $p_{old}$  has to be set equal to  $p$  and  $b_{old}$  equal to  $b$ .  $b_{ref}$  is the desired token bucket occupancy to which we want to regulate. Note that when marking out, no bytes are subtracted from the token bucket.

The stability of the token bucket occupancy  $b$  depends on the parameters  $k_1$  and  $k_2$ . Therefore, the appropriate choice of  $k_1$  and  $k_2$  is key to achieve the performance objective. Fig. 5 shows a block diagram of a linearized control loop to stabilize the token bucket occupancy  $b$ , on the basis of which  $k_1$  and  $k_2$  are computed according to the algorithm already described.

Fig. 6 depicts the token bucket occupancy  $b$  of customer C2 in case the data is transmitted according to the invented method, i.e. by using the random probability  $p$ . The token bucket occupancy stabilizes at a value of approx.  $b_{ref} = 0.75$  B. The aggregate traffic obtained by customer C2 in this case is 9.65 Mbps, which is much closer to the assigned maximum

bandwidth than the 8.32 Mbps obtained in the first simulation. Note that in Fig. 6, as compared to Fig. 3, the time intervals over which the token bucket is full are considerably shorter.

The objective of providing a throughput as close as possible to the assigned maximum bandwidth CIR can be reformulated as stabilizing the token bucket occupancy  $b$  around a reference value  $b_{ref}$ . In this specific exemplary embodiment  $b_{ref} \approx 0.75 B$ . A constant not full token bucket occupancy  $b$  implies a sending rate of  $in$  packets approximately equal to the assigned maximum bandwidth CIR. Since  $in$  packets are very unlikely to be dropped, this leads to a throughput approximately equal to the assigned maximum bandwidth CIR.

The method according to the present invention therefore relies on early notifying TCP sources of upcoming congestion via  $out$  marking based on the random probability  $p$ . In this way, the method according to the invention avoids synchronization among the TCP sources of C1 and C2, respectively, resulting in a better utilization of the contracted throughput and a better distribution of the total bandwidth in case customer C1 and C2 have contracted different CIRs. This results in a high level of fairness between customers C1 and C2.

One of the main advantage of the method according to the present invention is its simplicity. Instead of keeping state of each active connection, the method according to the invention only requires a small number of additional fixed and variable parameters for each token bucket. Another specific advantage is that its configuration does not require specific knowledge about the customer's traffic, but only a lower bound for the number of TCP sessions and an upper bound for the round trip time (RTT).

In the following we describe some simulation scenarios and their results to further explain the teachings according to the present invention. We continue assuming a DiffServ environment using a token bucket and a WRED queue. Such a scenario has proven very efficient in providing the agreed CIR in many simulated scenarios.

However, we observed that in a number of cases of interest in practice, such an architecture is not able to contrast fairness problems due to the TCP flow control mechanism. By employing the random probability, results can be significantly improved. In the present simulation scenarios, the discarding thresholds for conforming traffic in the WRED mechanism are set to a value that avoids in packets drops. Besides, the maximum threshold for packets marked out,  $OUT_{max}$  is equal to 10. Finally, for simulations of the method according to present invention, the instantaneous queue length for the AQM mechanism is taken into account, so that the system reacts faster to the early marking. Simulations were run using ns-2.

In the following we first show some simulation results we obtained by considering a number of heterogeneous scenarios. In the first three scenarios we assumed a fully subscribed link, i.e. the sum of the CIRs is equal to the bottleneck capacity. In contrast, in the fourth scenario we explored the behavior when the link is only partially subscribed. We conclude the section by evaluating the performance of the proposed marking scheme as a function of different parameters. All simulations were run using TCP Reno.

The first scenario is depicted in Fig. 7 and described by Table 1. The access links do not introduce either delays or

packet drops. It is known that non-responsive User Datagram Protocol (UDP) traffic causes problems of fairness when interacting with TCP flows. Therefore, in this scenario we study the interaction between customers transmitting either TCP only flows or mixed TCP and UDP traffic. To model UDP traffic, we considered Constant Bit Rate (CBR) flows, each sending at 1.5 Mbps. In this case the UDP rate sums up to 75% of the agreed CIR.

	CIR (Mbps)	# of flows		RTT (ms)	no p (Mbps)	p (Mbps)
		TCP	UDP			
Total	-	-	-	-	37.88	39.47
C1	10	10	0	20	9.46	10.10
C2	10	10	0	100	7.99	9.05
C3	10	10	5	20	10.35	10.21
C4	10	10	5	100	10.06	10.09

TABLE 1

Table 1 reports some settings we selected for this test and in the last two columns it shows the results in terms of traffic for the standard method and the method according to the present invention respectively. Table 1 also shows that using the random probability  $p$  helps customers sending TCP flows only to receive a higher share of the total bandwidth. In particular C1, characterized by a small RTT, achieves the agreed CIR while C2 gets more than 90% of it, against the 80% allowed by the standard method.

In a second scenario we assume heterogeneous values for the maximum assigned bandwidth CIR. A fairness problem also arises when different customers contract heterogeneous values for the assigned maximum bandwidth CIR. In fact, those customers characterized by a lower CIR value are

favorable in achieving the agreed CIR. The following scenario is an example of this behavior. The bottleneck link speed is set equal to 22 Mbps. Table 2 shows that in the considered case, the method according to the present invention allows to improve the overall link utilization by more than 15% and above all it leads to a significantly more fair bandwidth distribution.

	CIR (Mbps)	# of flows TCP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	18.18	21.62
C1	10	10	20	8.63	10.16
C2	10	10	100	7.07	9.23
C3	1	10	20	1.43	1.16
C4	1	10	100	1.03	1.06

TABLE 2

In a third simulation scenario we investigate the influence of the number of customers. When the number of customers and of flows grows to high values, then the multiplexing gain has a positive effect towards better link utilization and bandwidth distribution, even when the standard token bucket is used. The bottleneck link speed is set equal to 100 Mbps. In Table 3 we show simulation results that confirm this. However, also in this case, the method according to the present invention slightly improves the overall performance.

	CIR (Mbps)	# of flows TCP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	97.17	98.63
C1	10	40	20	10.36	10.58
C2	10	10	100	9.16	9.25
C3	10	10	20	9.91	10.10
C4	10	40	100	10.11	10.27
C5	10	20	20	10.20	10.33
C6	10	20	100	9.79	9.89
C7	10	15	20	10.11	10.25
C8	10	15	100	9.47	9.66
C9	10	5	20	8.88	9.05
C10	10	10	100	9.14	9.22

TABLE 3

In a fourth simulation, we investigate the interaction among customers with only TCP flows or only UDP flows respectively in an under-subscribed link. We considered a link speed of 53 Mbps, while  $\sum_{i=1}^4 CIR_i = 40 \text{ Mbps}$  (75% subscribed link). C3 and C4 transmit both 10 CBR flows, each at a rate of 1.5 Mbps, i.e. their sending rate is slightly above the CIR.

	CIR (Mbps)	# of flows TCP + UDP	RTT (ms)	no p (Mbps)	p (Mbps)
Total	-	-	-	49.56	51.31
C1	10	10+0	20	11.34	14.30
C2	10	10+0	100	9.72	10.48
C3	10	0+10	20	14.25	13.44
C4	10	0+10	100	14.24	13.08

TABLE 4

Table 4 shows that the method according to the present invention allows TCP to obtain a significantly higher share of the excess bandwidth as compared to the standard approach. This is especially true for more aggressive TCP customers such as C1, which has a smaller RTT, while C2 having a relatively small number of data flows and a large RTT (respectively 10 and 100 ms) can only achieve the assigned maximum bandwidth CIR.

In the following we investigate the benefit offered by the method according to the present invention as a function of the number of customers. To this end, we considered again the setting implemented for the third scenario. We evaluated the throughput achieved respectively by C1 and C2 as a function of the total number of customers. Customer C1 is characterized by a low RTT and a large number of data flows, therefore it is very likely that he will achieve the assigned maximum bandwidth CIR. Customer C2 on the contrary has a large RTT and a relatively small number of flows, thus he is penalized in the bandwidth sharing. In this simulation, we always considered the first n customers in Table 3 for a scenario with n customers.

In Fig. 8 we compare the throughput obtained by C1 and C2 when using the method according to the present invention and a standard token bucket. The method according to the present invention always allows to achieve the best performance. However, the most significant improvement is achieved by customer C2 when the total number of customers is below 8. By employing the method according to the present invention, customer C2 always obtains at least 90% of the assigned maximum bandwidth CIR, while the standard token bucket considerably penalizes it when the total number of customers

is low. The latter case is generally common for ISP access links.

In addition, we also evaluated the total link utilization for the third simulation. The results are reported in Fig. 9. The improvement due to the method according to the present invention is considerable.

We now consider the effect of a low number of flows per customer, of the order of a few units (for instance home users). In particular we analyze the performance of a scenario with 10 customers, each transmitting 10 flows, except for one customer that sends a smaller number of flows. All customers are assigned a CIR of 10 Mbps, the RTT varies for the different customers between 20 and 100 ms and the bottleneck speed link is equal to 100 Mbps.

For the customer sending a small number of flows we evaluate the achieved throughput as a function of the number of flows, when employing the method according to the present invention as compared to the standard token bucket. Results are reported in Fig. 10. As expected, when the number of flows is small, the throughput obtained is significantly lower than the assigned maximum bandwidth CIR. However by using the method according to the present invention we observe a relevant improvement. In this simulation, by transmitting 5 flows, the customer already obtains the assigned maximum bandwidth CIR, while when no early marking is applied, the throughput achieved is still 10% lower than the assigned maximum bandwidth CIR.

With regard to additional advantageous embodiments of the teaching according to the invention, in order to avoid repetition, please refer to the general section of the

description as well as to the claims at the end of this document.

Finally, we would like to point out explicitly that the exemplary embodiments described above only serve to describe the teaching claimed by are not limited to the exemplary embodiments.

#### **4. Brief Description of Drawings**

Fig. 1 in a scheme, an exemplary embodiment of a simulation of the transmission of data according to a known method and the invented method,

Fig. 2 a schematic depiction of the known token bucket algorithm,

Fig. 3 the evolution of the token bucket occupancy in a transmission according to a known method without a random probability,

Fig. 4 a schematic flow diagram showing the marking of the packets according to the invented method,

Fig. 5 a schematic block diagram of a linearized control loop to stabilize the token bucket occupancy,

Fig. 6 the evolution of the token bucket occupancy in a transmission of data according to the invented method,

Fig. 7 in a scheme, an additional exemplary embodiment of a simulation of the transmission of data according to known methods and the invented method,

Fig. 8 the evolution of the achieved throughput as a function of the number of customers when using a known method as compared to the invented method,

Fig. 9 the evolution of the total achieved link utilization as a function of the number of customers when using a known method as compared to the invented method and

Fig. 10 the evolution of the achieved throughput as a function of the number of TCP data flows when using a known method as compared to the invented method.

#### **List of Reference Characters and Definitions**

b	token bucket occupancy
$b_{old}$	old token bucket occupancy
$b_{ref}$	value to which the token bucket occupancy is to be regulated
B	token bucket size
CIR	assigned maximum bandwidth
C1, C2 ... C10	customers (senders)
D1, D2	customers (recipients)
IN	state of low drop precedence
K	gain in the controller
L	packet length
N	number of TCP sources
OUT	state of high drop precedence
p	probability
$p_{old}$	old probability
R, RTT	round trip time
u	evenly distributed random number
W	TCP window size
z	zero of controller
$\omega_g$	maximum frequency

AF	Assured Forwarding
AQM	Active Queue Management
CBR	Constant Bit Rate
DiffServ	Differentiated Services
DSCP	Differentiated Services Codepoint
IP	Internet Protocol
ISP	Internet Service Provider
PHB	Per Hop Behavior
QoS	Quality of Service
TCP	Transmission Control Protocol
WRED	Weighted Random Early Detection

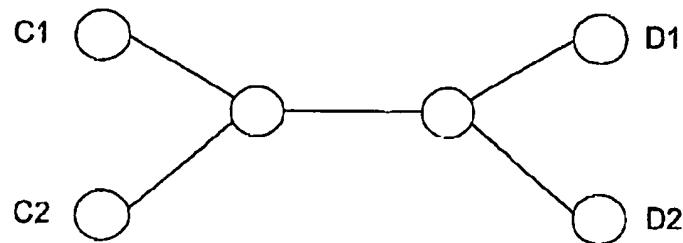


Figure 1

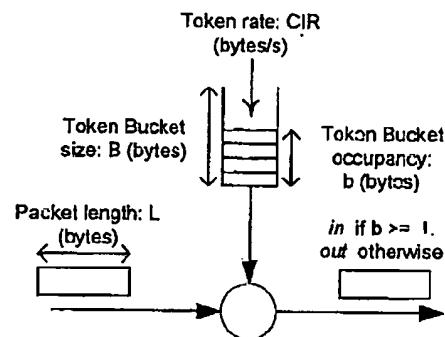


Figure 2

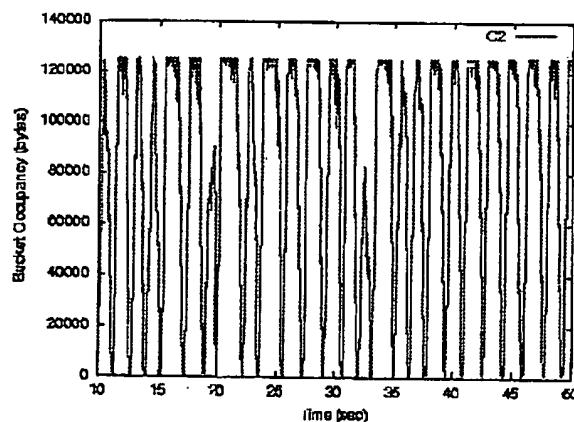


Figure 3

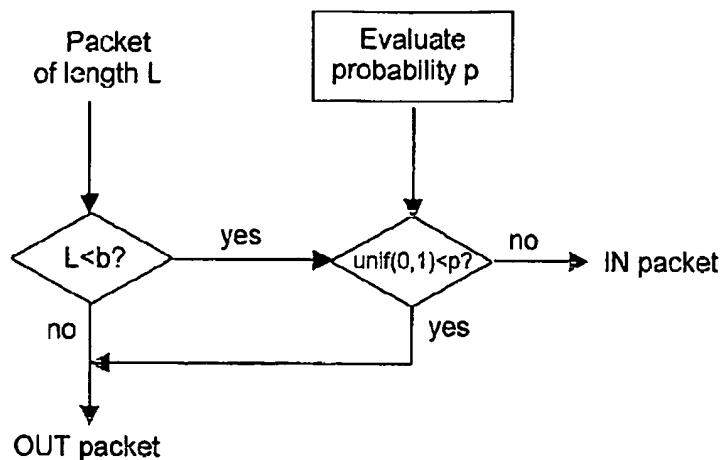


Figure 4

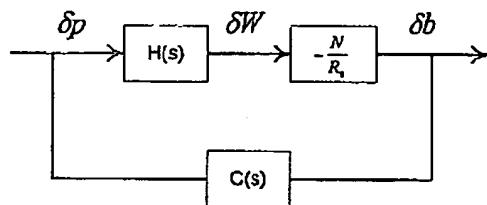


Figure 5

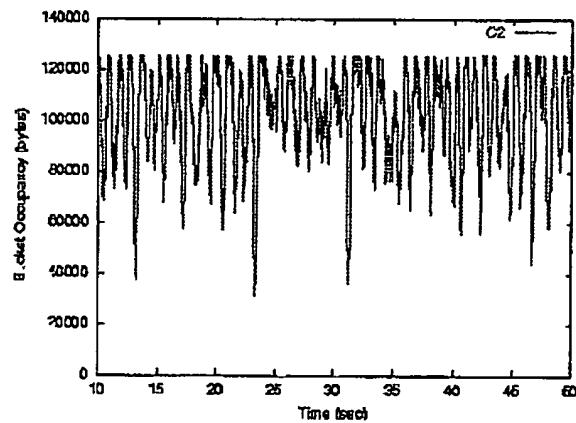


Figure 6

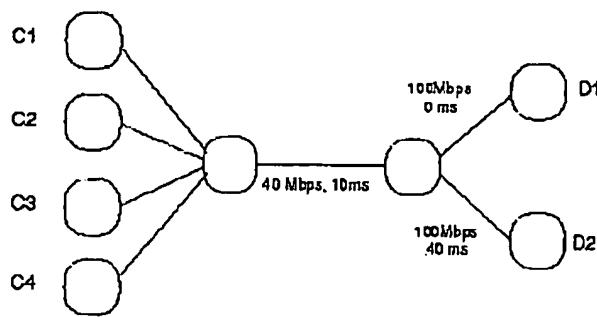


Figure 7

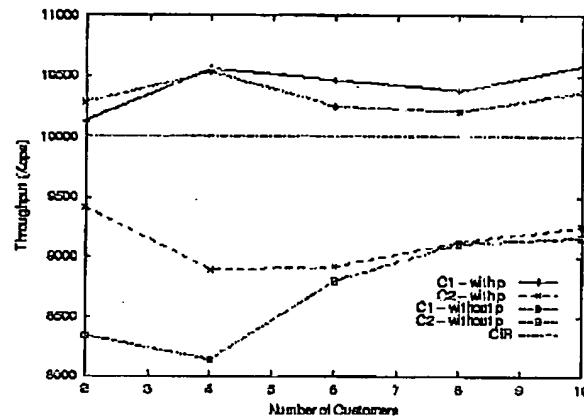


Figure 8

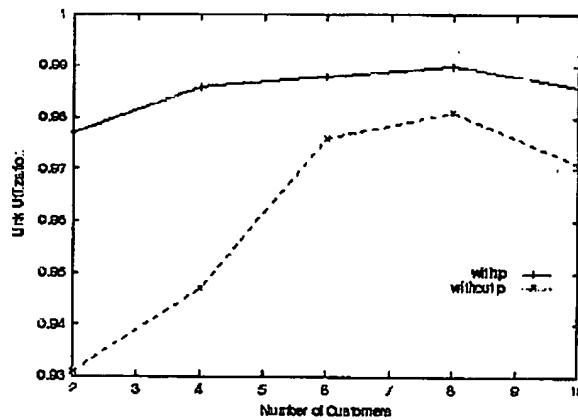


Figure 9

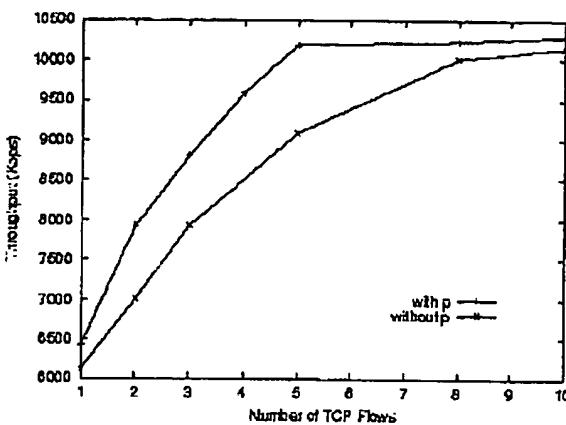


Figure 10

### 1. Abstract

A method of transmitting data from customers (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10) over a computer network, in particular over the Internet, where the data to be sent is split into packets, in particular into IP packets, where each packet is marked by one of at least two states (IN, OUT) and where the states (IN, OUT) determine which packets are dropped first, if packets are dropped during transmission, is, with regard to optimizing the drop rate of the packets, characterized in that the marking of the packet with a state of high drop precedence (OUT) is based on a random probability ( $p$ ).

### 2. Representative Drawing

Fig. 4

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record.**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**